

CSE 6740 Lecture 24

How Do I Optimize With Constraints? (Constrained Optimization)

Alexander Gray

agray@cc.gatech.edu

Georgia Institute of Technology

- ① Unconstrained Optimization: Latent-Variable
- ② Constrained Optimization

Unconstrained Optimization: Latent-Variable

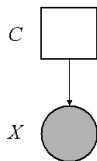
The EM algorithm, a form of bound optimization.

Mixture of Gaussians

Recall the mixture of Gaussians model, whose “hidden” variable is the class label:

$$P(C = k) = \pi_k, \quad \sum_k \pi_k = 1 \quad (1)$$

$$f(X|C = k) = N(\mu_k, \Sigma_k^2) \quad (2)$$



Mixture of Gaussians

Recall the mixture of Gaussians model, whose “hidden” variable is the class label:

$$P(C = k) = \pi_k, \quad \sum_k \pi_k = 1 \quad (3)$$

$$f(X|C = k) = N(\mu_k, \Sigma_k^2) \quad (4)$$

$$f(X) = \sum_{k=1}^K f(X|C = k)P(C = k) = \sum_{k=1}^K \pi_k N(\mu_k, \Sigma_k^2) \quad (5)$$

Mixture of Gaussians

Recall Bayes rule, which gives

$$P(C = k|x) = \frac{f(x|C = k)P(C = k)}{f(x)}. \quad (6)$$

This value is the probability that a particular component k was responsible for generating the point x , and satisfies $\sum_{k=1}^K P(C = k|x) = 1$. We'll use as a shorthand

$$w_{ik} \equiv P(C = k|x_i). \quad (7)$$

Mixture of Gaussians

We'll consider a simplified case where the covariances are fixed to be diagonal with all dimensions equal, $\Sigma_k = \sigma_k^2 I$, so

$$f(x|C = k) = N(\mu_k, \Sigma_k) = \frac{1}{(2\pi\sigma_k^2)^{D/2}} \exp\left\{-\frac{\|x - \mu_k\|^2}{2\sigma_k^2}\right\} \quad (8)$$

and

$$f(x) = \sum_{k=1}^K \pi_k \frac{1}{(2\pi\sigma_k^2)^{D/2}} \exp\left\{-\frac{\|x - \mu_k\|^2}{2\sigma_k^2}\right\}. \quad (9)$$

Maximum Likelihood, Identifiability

We want to find the parameters $\theta = \{\pi_k, \mu_k, \sigma_k\}$ which maximize the likelihood

$$L(\theta) = \prod_{i=1}^N f_{\theta}(X_i), \quad (10)$$

or $L(\theta|x) = f(x|\theta)$.

Unfortunately there exist parameter settings for which the likelihood goes to infinity, for example when one of the Gaussian components collapses onto one of the data points. Also there may be several parameter settings with identical likelihoods. We'll just ignore all this and proceed, because it turns out to be fine in practice.

Minimizing the Negative Log-likelihood

It is equivalent to minimize the negative log-likelihood

$$E \equiv -\log L(\theta) = -\sum_{i=1}^N \log f_{\theta}(X_i) \quad (11)$$

$$= -\sum_{i=1}^N \log \left(\sum_{k=1}^K f(X_i|C = k)P(C = k) \right) \quad (12)$$

Since this error function is a smooth differentiable function of the parameters, we can employ its derivatives to perform unconstrained optimization on it.

Minimizing the Negative Log-likelihood

For the centers μ_k we obtain

$$\frac{\partial E}{\partial \mu_k} = \sum_{i=1}^N w_{ik} \frac{(\mu_k - x_i)}{\sigma_k^2} \quad (13)$$

and for the σ_k we obtain

$$\frac{\partial E}{\partial \sigma_k} = \sum_{i=1}^N w_{ik} \left(\frac{D}{\sigma_k} - \frac{\|x_i - \mu_k\|^2}{\sigma_k^3} \right). \quad (14)$$

Minimizing the Negative Log-likelihood

Optimizing for the mixing parameters π_k must be done subject to the constraints

$$\sum_{k=1}^K \pi_k = 1, \quad (15)$$

$$0 \leq \pi_k \leq 1. \quad (16)$$

Minimizing the Negative Log-likelihood

This can be done by representing the mixing parameters in terms of a set of auxiliary variables γ_k such that

$$\pi_k = \frac{\exp(\gamma_k)}{\sum_{k=1}^K \exp(\gamma_k)}. \quad (17)$$

Recall that this is the logistic or softmax transformation. It ensures for $-\infty \leq \gamma_k \leq \infty$ that the constraints on π_k hold.

Minimizing the Negative Log-likelihood

Utilizing

$$\frac{\partial \pi_j}{\partial \gamma_k} = \pi_k I(j = k) - \pi_k \pi_j \quad (18)$$

and the chain rule consequence

$$\frac{\partial E}{\partial \gamma_k} = \sum_{j=1}^K \frac{\partial E}{\partial \pi_j} \frac{\partial \pi_j}{\partial \pi_k} \quad (19)$$

we can obtain

$$\frac{\partial E}{\partial \gamma_k} = - \sum_{i=1}^N (w_{ik} - \pi_k). \quad (20)$$

Minimizing the Negative Log-likelihood

Note that, at this point we are armed with derivatives, so we can use standard optimizers.

The EM algorithm does not actually require these derivatives in its final form (though we will consider these derivatives in our derivation which gets us there).

Conditions at the Optimum

It is insightful to see what the maximum likelihood solutions look like; when these derivatives are zero we have

$$\hat{\mu}_k = \frac{\sum_i w_{ik} x_i}{\sum_i w_{ik}} \quad (21)$$

$$\hat{\sigma}_k^2 = \frac{1}{D} \frac{\sum_i w_{ik} \|x_i - \mu_k\|^2}{\sum_i w_{ik}} \quad (22)$$

$$\hat{\pi}_k = \frac{1}{N} \sum_i w_{ik} \quad (23)$$

which represents the intuitively satisfying result that they are the usual mean and standard deviation where the points are weighted by the posterior probabilities of being generated by each component.

EM Algorithm for Mixture of Gaussians

These are the final update equations:

$$\mu_k^{\text{new}} = \frac{\sum_i w_{ik}^{\text{old}} x_i}{\sum_i w_{ik}^{\text{old}}} \quad (24)$$

$$(\sigma_k^2)^{\text{new}} = \frac{1}{D} \frac{\sum_i w_{ik}^{\text{old}} \|x_i - \mu_k^{\text{new}}\|^2}{\sum_i w_{ik}^{\text{old}}} \quad (25)$$

$$\pi_k^{\text{new}} = \frac{1}{N} \sum_i w_{ik}^{\text{old}}. \quad (26)$$

EM: Recurrence Idea

We can write the change in error in the form

$$E^{\text{new}} - E^{\text{old}} = - \sum_i \log \left(\frac{f^{\text{new}}(x_i)}{f^{\text{old}}(x_i)} \right) \quad (27)$$

$$= - \sum_i \log \left(\frac{\sum_k f^{\text{new}}(x_i | C = k) P^{\text{new}}(C = k)}{f^{\text{old}}(x_i)} \frac{P^{\text{old}}(C = k | x_i)}{P^{\text{old}}(C = k | x_i)} \right) \quad (28)$$

where the last factor is simply the identity.

EM: Bounding Idea

To proceed, we make use of *Jensen's inequality*, which states that, given a set of numbers $\lambda_k \geq 0$ such that $\sum_k \lambda_k = 1$, for any numbers z_k

$$\log \left(\sum_k \lambda_k z_k \right) \geq \sum_k \lambda_k \log(z_k). \quad (29)$$

Since the probabilities $P^{\text{old}}(C = k|x)$ in the numerator have these properties, they can play the role of the λ_k . This yields

$$E^{\text{new}} - E^{\text{old}} \leq (30)$$

$$- \sum_i \sum_k P^{\text{old}}(C = k|x_i) \log \left(\frac{f^{\text{new}}(x_i|C = k)P^{\text{new}}(C = k)}{P^{\text{old}}(C = k|x_i)f^{\text{old}}(x_i)} \right). \quad (31)$$

EM: Bound Minimization

We wish to minimize E^{new} with respect to the “new” parameters. If we let Q be the right-hand side of the previous equation, then we have

$$E^{\text{new}} \leq E^{\text{old}} + Q \quad (32)$$

and so $E^{\text{old}} + Q$ represents an upper bound on the value of E^{new} . We can therefore seek to minimize this bound with respect to the “new” values of the parameters.

By construction, this will necessarily lead to a decrease in the value of E^{new} unless E^{new} is already at a local minimum.

EM: Bound Minimization

If we now drop terms which depend only on the “old” parameters, we can replace Q by

$$\tilde{Q} = - \sum_i \sum_k P^{\text{old}}(C = k | x_i) \log(f^{\text{new}}(x_i | C = k) P^{\text{new}}(C = k)). \quad (33)$$

The smallest value for the upper bound is found by minimizing \tilde{Q} with respect to the “new” parameters, by finding the derivatives for them and setting them to zero.

For the μ_k and σ_k this is straightforward, but for the mixing parameters π_k we again have to account for the constraint $\sum_k P^{\text{new}}(C = k) = 1$.

EM: Bound Minimization

We can do this by introducing a Lagrange multiplier λ and minimizing the function

$$\tilde{Q} + \lambda \left(\sum_k P^{\text{new}}(C = k) - 1 \right). \quad (34)$$

Setting the derivative of $P^{\text{new}}(C = k)$ to zero we obtain

$$-\sum_i \frac{P^{\text{old}}(C = k|x_i)}{P^{\text{new}}(C = k)} + \lambda = 0. \quad (35)$$

After some manipulation of this, it can be found that $\lambda = N$.

EM Algorithm for Mixture of Gaussians

We arrive at the following update equations:

$$\mu_k^{\text{new}} = \frac{\sum_i w_{ik}^{\text{old}} x_i}{\sum_i w_{ik}^{\text{old}}} \quad (36)$$

$$(\sigma_k^2)^{\text{new}} = \frac{1}{D} \frac{\sum_i w_{ik}^{\text{old}} \|x_i - \mu_k^{\text{new}}\|^2}{\sum_i w_{ik}^{\text{old}}} \quad (37)$$

$$\pi_k^{\text{new}} = \frac{1}{N} \sum_i w_{ik}^{\text{old}}. \quad (38)$$

The “expectation step” (*E-step*) consists of evaluating the conditional probabilities w_{ik} using the last values of the parameters. The “maximization step” (*M-step*) consists of the updates to the parameters which move toward the local maximum.

EM Improvements

EM is based on first derivatives and has linear convergence. It is good away from the optimum but slow near it.

We can use the idea of *deterministic annealing*, analogous to simulated annealing, to effectively smooth out the likelihood surface so that the steps are only sensitive to major wells.

We can also make a hybrid method which switches to line search near the minimum.

EM Generalizations

A *generalized EM algorithm* simply takes a step which improves the likelihood but doesn't necessarily maximize the Q function.

The EM algorithm is an instance of a more general idea variously called *optimization transfer*, *bound optimization*, *block relaxation methods*, and *iterative majorization*.

Convex Optimization Problems

Problems with a convex objective function and, if there are constraints, convex constraints.

Linear Programming

When the objective and constraint functions are all affine, the problem is called a *linear program* (LP), which has the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} c^T x + d \quad (39)$$

$$\text{subject to } \quad Gx \leq h \quad (40)$$

$$Ax = b. \quad (41)$$

Quadratic Programming

When the objective function is quadratic and the constraint functions are affine, the problem is called a *quadratic program* (QP), which has the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} \frac{1}{2} x^T P x + q^T x + r \quad (42)$$

$$\text{subject to} \quad Gx \leq h \quad (43)$$

$$Ax = b. \quad (44)$$

Quadratically Constrained Quadratic Programming

If the constraints are also quadratic, the problem is called a *quadratically constrained quadratic program* (QCQP):

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} \frac{1}{2} x^T P x + q^T x + r \quad (45)$$

$$\text{subject to } \frac{1}{2} x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, M \quad (46)$$

$$Ax = b. \quad (47)$$

Second-order Cone Programming

A closely related problem is called a *second-order cone program* (SOCP), which has the form

$$\text{Find} \quad x^* = \arg \min_{x \in \mathbb{R}^D} f^T x \quad (48)$$

$$\text{subject to} \quad \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \dots, M \quad (49)$$

$$F x = g. \quad (50)$$

A constraint of this form is called a second-order cone constraint.

Geometric Programming

A *geometric program* (GP) is a problem of the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) \quad (51)$$

$$\text{subject to } c_i(x) \leq 1, \quad i = 1, \dots, M \quad (52)$$

$$d_i(x) = 1, \quad i = 1, \dots, N \quad (53)$$

$$(54)$$

where f and the c_i have the form

$$\log \left(\sum_k e^{a_{1k}^T x + b_{1k}} \right) \quad (55)$$

and the d_i have the form $e^{g_i^T x + h_i}$.

Semidefinite Programming

A *semidefinite program* (SDP) has the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} c^T x \quad (56)$$

$$\text{subject to } x_1 F_1 + \dots + x_n F_n + G \leq 0 \quad (57)$$

$$Ax = b. \quad (58)$$

Convex Optimization Methods

The interior-point method.

Lagrangian Duality

Consider again the general form for an optimization problem:

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) \quad (59)$$

$$\text{subject to } c_i(x) \geq 0, \quad i = 1, \dots, M \quad (60)$$

$$d_i(x) = 0, \quad i = 1, \dots, N. \quad (61)$$

Lagrangian Duality

We can rewrite the original problem as the unconstrained optimization problem

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) + \sum_i^M l_\infty(c_i(x)) + \sum_i^N l_\infty(d_i(x)). \quad (62)$$

where l_∞ is the indicator-like function which takes the value 0 if the constraint function is satisfied by x and ∞ otherwise.

Lagrangian Duality

The basic idea of Lagrangian duality is to soften the problem, by replacing $I_\infty(c_i(x))$ by $\lambda_i c_i(x)$ and $I_\infty(d_i(x))$ by $\eta_i d_i(x)$, where the λ_i and η_i are positive weights, obtain the *Lagrangian* of the problem:

$$L(x, \lambda, \eta) = f(x) + \sum_i^M \lambda_i c_i(x) + \sum_i^N \eta_i d_i(x). \quad (63)$$

The λ_i and η_i are called the *Lagrange multipliers*, and the λ and η vectors are called the *dual variables* or *Lagrange multiplier vectors* associated with the problem.

Lagrangian Duality

We define the *dual function* g as the minimum value of the Lagrangian over x :

$$g(\lambda, \eta) = \inf_x L(x, \lambda, \eta). \quad (64)$$

Since the dual function is the pointwise infimum of a family of affine functions of (λ, η) , it is concave, even if the original problem is not convex. The dual function yields a lower bound on the optimal value: $g(\lambda, \eta) \leq x^*$.

Lagrangian Duality

The dual function value $g(\lambda, \eta)$ is its optimal value over x . Now we'd like to find the best lower bound that can be obtained from the dual function:

$$\text{Find } \lambda^*, \eta^* = \arg \max_{\lambda, \eta} g(\lambda, \eta) \quad (65)$$

$$\text{subject to } \lambda \geq 0. \quad (66)$$

This is called the *dual problem*, while the original problem is called the *primal problem*. This is always a convex optimization problem because the objective function is concave and the constraint is convex, even if the primal problem is not convex.

Lagrangian Duality

If the primal problem is convex, usually maximizing the dual problem is the same as minimizing the primal problem. We call this *strong duality*.

Suppose we have strong duality, and all the functions are differentiable. Since x^* minimizes $L(x, \lambda^*, \eta^*)$ over x , it follows that its gradient is zero at x^* :

$$\nabla f(x^*) + \sum_i^M \lambda_i^* \nabla c_i(x^*) + \sum_i^N \eta_i^* \nabla d_i(x^*) = 0. \quad (67)$$

Optimality Conditions

Thus we have these conditions which must hold at the optimum:

$$c_i(x^*) \leq 0 \quad (68)$$

$$d_i(x^*) = 0 \quad (69)$$

$$\lambda_i^* \geq 0 \quad (70)$$

$$\lambda_i^* c_i(x^*) = 0 \quad (71)$$

$$\nabla f(x^*) + \sum_i^M \lambda_i^* \nabla c_i(x^*) + \sum_i^N \eta_i^* \nabla d_i(x^*) = 0. \quad (72)$$

These are called the *Karush-Kuhn-Tucker* (KKT) conditions. They are necessary and sufficient at the optimum. We can thus formulate optimization as solving these equations.

Logarithmic Barrier Idea

We'll now do something related but slightly different with the constraints. We rewrite the general problem as

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) + \sum_i^M I_\infty(c_i(x)) \quad (73)$$

$$\text{subject to } Ax = b. \quad (74)$$

We approximate the indicator function by

$$\hat{I}(u) = -\frac{1}{t} \log(-u) \quad (75)$$

where t is a parameter which increases the accuracy of the approximation as it increases. $\hat{I}(u)$ goes to ∞ as u increases to zero, but is differentiable, and convex.

Logarithmic Barrier Idea

We call the function

$$\phi(x) = - \sum_i^M \log(-c_i(x)) \quad (76)$$

the *logarithmic barrier* for the problem. We'll optimize $f(x) + \frac{1}{t}\phi(x)$.

The *barrier method* solves a sequence of such problems (each of which is convex), increasing t on each iteration, using Newton's method. The solution $x^*(t)$ is the starting point for the next value of t .

It can be shown that the error for each iteration is bounded by M/t , and thus the error goes to zero.

Interior-Point Method

The barrier method is an example of an *interior-point* method.

Given feasible x , $t > 0$, $\mu > 0$, tolerance $\epsilon > 0$, repeat:

- 1 Find $x^*(t)$ by using Newton's method to minimize $tf + \phi$ subject to $Ax = b$, starting at x .
- 2 $x = x^*(t)$.
- 3 Quit if $M/t < \epsilon$.
- 4 $t = \mu t$.

Methods called *phase I* methods are used to choose the starting x .

Interior-Point Method

The interior-point method can be used for all of the constrained convex optimization problems.

In practice, despite convergence analysis which relates the number of iterations to M , it always takes about 10-20 iterations.

There is a modification of the barrier method called the *primal-dual* interior-point method, which is often a bit faster, and is what is used in practice.

SVM Quadratic Program

Recall that the formulation of the support vector machine results in a quadratic program.

Though generally effective, interior-point is not sufficient to handle the large number of variables and constraints in an SVM problem. An idea called *chunking* is often used for large-scale convex optimization problems, which solves smaller problems using subsets of the constraints, while adding more constraints until all of them are satisfied.

SMO: Extreme Chunking

The *sequential minimization optimization* algorithm was developed specifically for SVM's. It is a special case of chunking which considers only two constraints at a time, the minimum number.

While it can be shown to converge, it is based on several heuristic steps. Empirically it is much more efficient than interior-point or general chunking.

The exact reason it has an advantage in this setting has not yet been elucidated.

Main Things You Should Know

- Interpretation of the EM algorithm in terms of bound optimization
- The different types of constrained optimization problems
- The idea of the interior point method